COMAL Standards Meeting

by Brian Grainger
Introduction by TeleNova

[Editors note: everyone knows that we need a COMAL Standards Group - but since little is heard about them, stories like this can result. Reprinted from ICPUG, Vol 8 No 5.]

The story so far ...

In the (BASIC) beginning was the word, and the word was GOTO...

Now, Borge (Christensen) looked upon the word and saw that it was bad (or at least if not all bad, then certainly much overused). And Borge did address the multitude saying, "Behold! I have invented a new word, and the word is PROC!"

Now the people looked upon this word, and they saw that it was good. Thus they embraced the word, and built a new language to house it.

And it came to pass that a True COMAL Deity entered upon the scene and established a group of disciples who were all similar worshippers of the one True Language.

Now for many years this group did use up much company expenses in mutual back-slapping and (liquid) celebration of The Language.

But lo! They did not notice that the BASIC reason for The Language was disappearing because it had built up even more followers, who were not all living on expense accounts. And so it came to pass that the True COMAL Deity lost interest in the language, saying "whosoever wishes to lead the worship may do so, but there's nothing in it for me". And there was much wailing and gnashing of teeth, and the disciples spoke amongst themselves saying "why has our God forsaken us, and what's more, how are we going to preserve our biannual expense account jollies?"

And so the disciples travelled to the land of Bagpipes and Sporrans (and more importantly, Scotch) in order to debate these matters, hoping that it wouldn't be the last time, but making sure that if it was they made the most of it...

That is a humorous account of events up to the time of the 1985 COMAL Standardization meeting as written by representatives of TeleNova, a Swedish company. The rumours of Borge deserting COMALites were exaggerated somewhat, as were the thoughts that the 1985 meeting may have been the last. In August, the 1986 Standardization meeting was held in Denmark and I accepted the invitation to attend, representing users of COMAL in England. I hasten to add it was at my own expense.

I went to Denmark for the two and a half days of meetings, not sure of what to expect but with eager anticipation. At long last I was participating in how the language would develop.

There are two groups involved, the Standardization Group, and the Development Group. The latter consists of all those attending and the purpose is to discuss the various proposals that may be put forward for changes to COMAL. In order for the Development Group to accept a proposal, two thirds of members present must support it. Agreement on a proposal means that it can be carried forward to the Standardization Group, where the ultimate decision is made. The Standardization Group consists of implementors of COMAL only. Other attenders, such as myself, can only

More ►

observe at the Standardization Group meetings. We have no voting rights. For the Standardization Group to accept a proposal, four fifths of members present have to support it.

The majorities required to get proposals accepted may seem rather high but this way ensures the COMAL language does not get changed on a whim. Certainly there is no universal agreement among the implementors on how the language should develop. It is my feeling that because an almost universal level of support is required to accept a proposal, the user is protected. User programs do not become obsolete as the language develops due to lack of compatibility between one version of the language and the next.

In Denmark, during the three days, there was an attendance of around 20, from 5 countries, representing 6 implementors, and the users of England, Scotland, and Eire. In addition, the writers of the manual for the COMAL cartridge (*Cartridge Tutorial Binder*) put in an appearance, as did Borge himself.

Having only come into contact with one implementor previously, UniComal, who produced the COMAL cartridge for the C64, it was fascinating to meet some of the others. The views expressed varied widely. The very conservative wanted to make few changes to the language in case they made ANY existing program obsolete. On the other hand, the radicals wanted to make lots of changes so as to provide facilities in COMAL for any purpose imaginable! It is amazing that anything gets agreed at all, but the result of the meeting was to steer a course between the two extremes, with some additons being agreed and more to be put to a vote at a later time.

A word about the various implementors of COMAL who were present may be of interest to those who think that COMAL is limited to Commodore machines.

UniComal need no introduction. Of all the implementors, they have decided to go for the international market with versions for Commodore and IBM.

RegenCentralen (RC) who have implemented COMAL on the RC Piccolo, a machine much used in Danish Schools.

TeleNova who have implemented COMAL on Compis computers which are mainly used in Sweden.

Dansk Data Elektronik have implemented COMAL under the Unix operation system, but, again, seem to be limited to Denmark.

Dench Systems are a company working with UniComal developing COMAL in real-time applications.

Finally, and the biggest surprise to me, was Dalgasoft, who have COMAL implemented on the Amstrad 464 and 6128 and are working on a CP/M version.

John Clack, a user representative at the meetings, uses COMAL on a VAX.

Apart from the VAX version, all the versions were on display in what became known as the Hornstrup Computer Centre (Hornstrup being the place where the meeting took place). One does tend to get rather isolated in one's thinking in using only Commodore machines, and an important outcome of the meeting for me was to see other versions being used. The subtle differences could be weighed up, but the value of a standard and usefulness of the

More ►

group becomes clear when you can type exactly the same program into a Commodore 64, IBM PC, Amstrad 464, Piccolo, or Compis and they all work!

The meetings discussed 22 proposals, mostly for additions to the COMAL extensions. They ranged from the simple, such as introducing **a$:+b$** as a shorthand for **a$:=a$+b$** to the complex, such as the ability to address subsets of an array with one variable descriptor. One can guess how complex the discussions became by the results. The Development Group passed 8 of the proposals, having amended the syntax on 2 of the 8. It threw out 13 and liked the idea of the remaining one, but asked for the syntax to be reconsidered. The Standards Group then passed 2 of the 8 with the others going to a postal vote since the group had not had sufficient warning of the proposals.

So, what were the two agreed proposals that appear to be the only achievements of two and a half days?

(1) The ability to assign the same value to every element of an array with a single statement. Example:

```
a(,):=1
```

(2) The ability to PRINT or INPUT whole arrays by one statement. Example:

```
DIM a(10,10)
a(,):=1
PRINT a(,);
```

This would print the 100 elements of the two dimensional array (set to the value of 1) with a single space between each value.

In reality, the meeting achieved much more than this. Apart from the six items awaiting the outcome of a postal vote, important discussions took place on the future of COMAL regarding (a) packages; (b) data structures especially record structures and pointers; and (c) real time and concurrency. In the fullness of time, after careful consideration, these items will form the basis of future enhancements that will widen the usage of COMAL. In view of my enthusiasm for graphics, I was also roped into submitting a proposal for a defined graphics extension (*also see the preliminary Graphics Kernal report in COMAL Today #7*, page 8).

After all the discussions the only business to conclude was when and where to hold the next meeting. It is likely to be in the Spring of 1987 and, one hopes, in London.

I had a great three days (plus one rest day). At one point, I remember the incredible amount of time it took, and the amount of heat generated in discussing:

"What value should be returned by the expression A$ IN B$ when A$ is the empty string?"

Possible answers include:

(a) LEN(B$)+1
(b) 0
(c) any positive number
(d) undefined

The reasoning behind (a) is that the empty string is regarded as part of every string and therefore A$ IN B$ must be TRUE (a value not equal to 0) when A$ is empty.

The reasoning behind (b) is that it is hard to explain why the following will not work on an IBM PC if (a) is used:

More ►

```
DIM a$ OF 2
PRINT "Shall I reformat the harddisk?"
REPEAT
  a$=KEY$
UNTIL a$ IN "yYnN"
IF a$ IN "yY" THEN PASS "FORMAT C:"
```

(On an IBM PC, KEY$ returns an empty string when no key is pressed! On a C64, CHR$(0) is returned!)

Further Reference:

COMAL Implementations, COMAL Today #11, page 10
COMAL Standards Conference, COMAL Today #7, page 8 ■

PSALM 23

COMAL is my language; I shall not want.

It maketh me to structure my thoughts;
It sorteth and indenteth my program.

It RESTOREth my DATA;
It leadeth me in paths of readability for its own sake.

Yea, though I walk through the valley in the shadow of BASIC,
I will fear no GOTO:
for it is state-of-the-art;
its PROCs and its FUNCs they comfort me.

It prepareth an unassigned data space before me in the absence of my values:
It reporteth my errors at source, even if my stack runneth over.

Surely goodness and reliability shall follow me all the days of my life:
and I will dwell in its syntax forever.

Reprinted from ICPUG, Sept/Oct 1986. Collected by Brian Grainger from the 1986 COMAL Standardization meeting. ■

Extra Programs

CAPITOLS

Richard and Todd Shagott have concocted an educational game driver called *can'you'tell'me*. The program uses the file *dat.quiz'menu* which contains the names and data files of each available game. They included games to guess 50 U.S. capitols and 315 world capitols. The data for these games is in *dat.capitols* and *dat.world'cap*. To play the games:

RUN "can'you'tell'me"

The uniqueness of this program is that once it asks the question, it randomly begins to insert letters into the answer, one at a time after a short delay. It decrements the possible number of points available as each letter is inserted. All questions are randomly selected, and then removed from the selection pool only if answered correctly.

In actual play, the <*space*> bar interupts play for the purposes of entering a guess, pausing for a break, or quitting. The games are on *Today Disk #15*.

3D AIRPLANE REVISITED

Today Disk #15 contains updated versions (0.14 and 2.0) of the *airplane* programs by Herbert Denaci from *COMAL Today #13*. They are *1520-3d'airplane* and *chip.airplane*. The 2.0 version first draws the airplane on the hi-res screen. It then checks to see if the program is running on a c128 with Super Chip. If it is, then it draws the airplane on the ultra hi-res (640*200) screen (switch your monitor to 80 column mode to view the plane). ■